What is claimed is:

1. A method for providing a stand-alone testing environment for a test object functional element of a computer system, said test object functional element having a plurality of interfaces for coupling with other elements of said computer system, said interfaces being of a type which provide communication between functional elements and which employ a predetermined interface protocol for inter-processing communication whose mode of operation involves a shared memory such that information communicated through said interfaces is passed between said functional elements by a process of notifying the addressed functional element that information is ready and providing the addressed functional element with its location in said shared memory, said method comprising:

> providing a computerized dialog to enable a user to create an input data file for said test object functional element in a form for subsequently being stored in an identifiable location in said shared memory;

26

prompting a user for at least one functional element

interface task which has been previously developed

utilizing said stand-alone testing environment and

which is of a form compliant with said predetermined

interface protocol and which is stored with its

identifiable location in said shared memory;

starting said at least one functional interface task

utilizing said computer dialog created input data

file; and

monitoring said plurality of interfaces.

2.   The method of claim 1 further comprising starting a user

supplied application system task along with said at least one

functional interface task.

3.   The method of claim 1 further comprising displaying a

status window while running said at least one functional

interface task.

4. The method of claim 1 wherein said predetermined interface protocol is further of a type in which the location of information is distributed among a set of at least two hierarchical levels of a database formed in association with said shared memory, said hierarchical levels being organized by degree of generality of functional interface task information to be stored therein, and the method further comprises:

creating a test case generation file by providing the user with a corresponding set of task creation options related to said at least one functional interface task individually operative with a degree of generality of functional task information that is to be stored in a corresponding individual level of said set of at least two hierarchical levels of said database.

5. The method of claim 1 further comprising storing a unique interface file corresponding to said at least one functional element interface task.

6. The method of claim 1 further comprising storing said input data file in a user defined file such that said user defined

file may be viewed and edited outside of said stand alone

testing environment.

7. A method for testing a test object functional element of a

computer system with a stand- alone functional element test

tool, said test object functional element having at least one

interface for communicating with other functional elements of

said computer system, said at least one interface having a

predetermined interface protocol for inter-processing

communication, said method comprising:

creating an input data file for said test object functional

element by prompting a user for data format and

content compatible with said predetermined interface

protocol;

storing said input data file;

creating a test generation file by providing said user with

a plurality of task creation options whereby selected

task creation options are input into said test

generation file which is written in a predetermined

high level interface programmers' language adapted for

compilation into computer code executable statements

compatible with said predetermined protocol;

compiling said test generation file and said input data

file to produce a test case executable file in a

preferred programming language based on said selected

task creation options;

initiating a test utilizing said test case executable file

and said input data file for testing said test object

functional element and said at least one interface by

monitoring a status of said test; and

storing test result data related to said test.

8. The method of claim 7 wherein said step of creating a test generation file further comprises selecting test initiation features.

9. The method of claim 7 wherein said step of creating a test generation file further comprises providing for at least one user defined button.

10. The method of claim 9 wherein said at least one user defined button is user operable for said step of initiating said test.

11. The method of claim 7 wherein said predetermined interface protocol for inter-processing communication employs a mode of operation involving a memory shared among said test object and said other functional elements and in which information to be communicated through the interface is passed between functional elements by a process of notifying an addressed functional element that data is ready and providing the addressed functional element with a corresponding location in said shared memory, said interface protocol further being of a type in which a location of information is distributed among a set of at least two hierarchical levels of a database formed in association with

31

said shared memory, said hierarchical levels being organized by

degree of generality of functional interface task information,

and the method further comprises:

said step of providing the user with a plurality of task

options including providing at least one corresponding

set of options individually operative with a

corresponding individual level of said set of at least

two hierarchical levels of said database.

12. The method of claim 11 further comprising displaying said

input data to a user on a file viewer.

13. The method of claim 7 further comprising comparing said test

result data with expected results from said test object

functional element utilizing said input data file.

14. A system operative for testing performance validity and

accuracy of a test object functional element, said test object

functional element forming a portion of a computer system, said

test object functional element having a plurality of

communication interfaces with said test object functional

element constrained to be operatively responsive to a

predetermined interface communication protocol, said system

comprising:

a test case data file producing subsystem for facilitating

the production by a user of at least one file of test

case data, said test case data producing subsystem

being operative for identification of an input data

structure and to utilize said input data structure to

prompt a user for input values of said test case data,

said test case data producing subsystem being

operative to store said at least one file of test case

data;

a test case generation file producing subsystem for

facilitating the production by said user of a test

case generation file, said test case generation file

producing subsystem providing a plurality of user

interface task options to provide the user with a

choice among them in developing a test case generation

file of a selected at least one interface task of said

plurality of interface tasks, said selected at least

33

one interface task being for communication to said

test object functional element through a first

predetermined at least one communication interface;

and

a test case execution subsystem to effect operation of said

test object functional element based on said user

selected at least one interface task and said at least

one file of test case data, whereby said test case

execution subsystem enables said user to test said

test object functional element for validity and

accuracy of its operation by monitoring a second

predetermined at least one of the remaining

communication of interfaces of said plurality of

communication interfaces.

15. The system of claim 14 wherein:

said input data structure is utilized to prompt a user for

test case data being in a form cooperatively

associated with said predetermined interface

communication protocol to constrain said at least one

34

test case data file to be compatible with said

predetermined interface protocol;

said plurality of user interface task options provided by

said test case generation file producing subsystem

being in form cooperatively associated with said

predetermined interface communication protocol to

constrain said selected at least one interface task to

be written in a predetermined high level interface

programmers' language adapted for compilation into

computer code executable statements compatible with

said predetermined interface protocol; and

said operation of said test object functional element

effected by said test case execution subsystem

comprising said operation of said test object

functional element using a file of compiled executable

statements based upon said test case data and said

test case generation file.

35

16. The system of claim 14 and:

said interface communication protocol being a protocol for
inter-process communication of an application
interface task from said test object functional
element to at least one other functional element which
also forms a portion of said computer system;

said plurality of interfaces including a subsystem for
implementing said inter-process communication
interface protocol comprising a memory operatively
connected to said test object functional element and
to said at least one other functional element by an
arrangement whereby said functional elements share
said memory; and

said subsystem for implementing the inter-process
communication interface protocol employing a mode of
operation in which data to be communicated through an
interface is passed between functional elements by a
process of notifying the functional element to which
an application interface task is be communicated that

36

data is ready and providing the addressed functional

element with the location of the data in said shared

memory.

17. The system of claim 14 wherein said test case execution

subsystem is operable to effect operation of another test object

functional element simultaneously with operation of said test

object functional element.

18. The system of claim 17 wherein said test case execution

subsystem is operable to monitor said at least one interface

between said test object function element and said another test

object functional element.

19. The system of claim 16 and:

said shared memory being adapted to form a database having

a set of at least two hierarchical database levels

organized by degree of generality of interface task

information; and

37

said plurality of user interface task options provided by

said test case generation file producing subsystem

including providing at least one corresponding set of

options individually operative solely with a

corresponding level of said set of at least two

hierarchical levels of said database.

20. The system of claim 15 wherein said test case generation
file producing subsystem is operative to provide the user a
choice among a plurality test initiation events to cause the
test to be performed upon a selected test initiation event to
start flow of said test case data into said first functional
element.